



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/649,198	08/27/2003	Panagiotis Kougiouris	11389-033 (2000874)	5126
23483	7590	02/21/2008		
WILMERHALE/BOSTON			EXAMINER	
60 STATE STREET			NGUYEN, MAIKHANH	
BOSTON, MA 02109			ART UNIT	PAPER NUMBER
			2176	
			NOTIFICATION DATE	DELIVERY MODE
			02/21/2008	ELECTRONIC

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

michael.mathewson@wilmerhale.com  
teresa.carvalho@wilmerhale.com  
sharon.matthews@wilmerhale.com

## Office Action Summary

Application No.	KOUGOURIS ET AL.
Examiner MAIKHANH NGUYEN	Art Unit 2176

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS,  
WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) Responsive to communication(s) filed on 29 May 2007.  
2a) This action is FINAL.                    2b) This action is non-final.  
3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) Claim(s) 1-19 is/are pending in the application.  
4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.  
5) Claim(s) \_\_\_\_\_ is/are allowed.  
6) Claim(s) 1-19 is/are rejected.  
7) Claim(s) \_\_\_\_\_ is/are objected to.  
8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) The specification is objected to by the Examiner.  
10) The drawing(s) filed on \_\_\_\_\_ is/are: a) accepted or b) objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).  
11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
a) All    b) Some \* c) None of:  
1. Certified copies of the priority documents have been received.  
2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.  
3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- 1) Notice of References Cited (PTO-892)  
2) Notice of Draftsperson's Patent Drawing Review (PTO-948)  
3) Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_.  
4) Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_.  
5) Notice of Informal Patent Application  
6) Other: \_\_\_\_\_.

## DETAILED ACTION

1. This action is responsive to Amendment filed 05/29/2007.

Claims 1-19 are presented for examination. Claims 1 and 3 have been amended.

### Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

*(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.*

*This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).*

Claims 1-19 are rejected under 35 U.S.C.103(a) as being unpatentable over of **Strong** (US 6167523, filed 05/1997) in view of **Lee et al.** (US 6535883 B1).

#### As to claim 1:

Strong teaches a method for automatically validating text that is input to a client computer (*col. 4, lines 16-37*), the method comprising:

- at the client computer, processing a markup language file to receive text input, the markup language file comprising a description of a graphical user interface, the description comprising a GUI element enable to receive the text input (e.g., *performing validation ... of input data from electronic forms such as Hypertext Markup Language forms; col. 3, lines 7-47*), wherein the markup language file instantiating a validation manager (e.g., *the forms data validation and processing control program 255 is stored on the Web server 205. The handlers 260 associated with HTML forms to be processed; col. 4, lines 62-67*);
- in response to processing the markup language file , displaying the GUI on a display screen at the client computer (e.g., *display the form 280 to the client PC 200 user such that he or she may fill out the form; col. 5, line 62-col.6, line 14*);
- receiving text that is input to the GUI element enabled to receive text input (e.g., *entering input data into it various fields ... A submit button 325 is also provided; col. 6, lines 5-21 & see fig. 3A and the associated text*);
- in response to receiving the text that is input to the GUI element, sending a programmatic event by to the validation manager (e.g., *entry of data into the form 280, the form 280 is submitted by the client PC 200 user by clicking on the submit button 325... submitting the form causes data 290 from the form including input data entered into the form to be transmitted from the client PC 200 to the server*

*205 ... When the data 290 is received by the server 205 ...the form data validation and processing program 255 controls data validation, error reporting and processing of the input data ... after the data 290 from the form 280 is received in step 400, the form validation and processing program 255 determines whether the data includes a first registry key identifier in step 405. If a registry key identifier is not identified by the program 255, then in step 410, a general error message is sent to the client computer system 200. The error message may indicate a server error, for example, or inform the client PC 200 user that the form cannot be processed; col. 7, lines 5-41).*

- in response to receiving the a programmatic event at the validation manager , determining at the client computer whether the received text is valid text input (e.g., *The Handlers subkey 502 defines how to process input data once validation has been successfully performed ... a program that automatically configures the registry keys and subskeys based on user responses to predetermine questions ...The form data validation ... validates INPUT type fields in an HTML form ... a field that does not have a mandatory entry field argument associated with it can still considered valid even if the user does not enter any information into the field. Information that is entered into the field, however, may still have to meet specified requirements in order for the information to be considered valid; col. 7, line 60- col. 9, line 18); and*

- providing an indication that the received text is invalid if the received text is determined to be invalid (*e.g., For each field as it is evaluated, if the data in the field is invalid according to requirements specified in the one or more configuration registry keys, an error message corresponding to the field being evaluated is dynamically built and logged in an error log ... if there are entries in the error log, a detailed and specific error message is provided to the user identifying the specific field(s) that included invalid data; col.3, lines 36-47/ col. 6, lines 50-60 & fig. 7 and the associated text.*)

Strong does teach instantiating the validation manager. Strong, however, does not explicitly teach a markup language tag for instantiating a validation manager at the client computer and in response to processing the markup language file at the client computer, instantiating the validation manager in response to said processing the markup language file.

Lee teaches a markup language tag for instantiating a validation manager at the client computer and in response to processing the markup language file at the client computer, instantiating the validation manager in response to said processing the markup language file (*col.5, line 23- col.6, line 39*).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify Strong with Lee because it would have provided the

advantages including flexible and robust data validation, processing and error reporting for many different electronic forms in many different environments, as well as a reduction in potential security risks that may be posed by providing data validation and processing information within an electronic form.

**As to claim 2:**

Strong teaches the validation manager instantiating a validation component, wherein said instantiating a validation component comprises specifying the type associated with the GUI element; wherein said validation manager determining whether the text input at the GUI element is valid text input comprises the validation manager calling the validation component; wherein said validation manager calling the validation component comprises the validation manager specifying the text input to the GUI element; wherein the validation is operable to return a result value to the validation manager indicating whether the text input received to the GUI element is valid text for the type associated with the GUI element (*col.3, lines 11-47; col.4, line 63-col.5, line 22; col.7, lines 5-41; and col.8, lines 1-64*).

**As to claim 3:**

Strong teaches receiving text input to the GUI element is performed by a user of the application provide text input to the GUI element (*col. 5, line 62-col.6, line 21; col. 9, line 4-18 & see fig. 3 and the associated text*).

**As to claim 4:**

Strong teaches an application providing text input to the GUI element (*col. 3, lines 14-32; col.6, lines 1-30; col. 7, line 1-31 & see fig. 3A and the associated text*).

**As to claim 5:**

Strong discloses one or more attributes for specifying when text input to the GUI element should be validated; the validation manager component is operable to validate text input to the GUI element in accordance with the one or more attributes specifying when text input to the GUI element should be validated (*col.3, lines 11-47; col.4, line 63-col.5, line 22; col.7, lines 5-41; and col.8, lines 1-64*).

**As to claim 6:**

Strong teaches each of the one or more attributes for specifying when text input to the GUI element should be validated corresponds to at least one type of programmatic event; the step of said validation manager receiving a programmatic event comprises the validation manager ignoring the programmatic event if the programmatic event does not correspond to one of the attributes for specifying when text input to the GUI element should be validated (*col.7, lines 5-41; and col.8, lines 1-64*).

**As to claim 7:**

Strong teaches receiving, among other things, clicking on the GUI element (*see the clicking of a button icon discussion beginning at col.13, line 49*); wherein said validation

manager component receiving a programmatic event in response to said providing text input to the GUI element comprises the validation manager component receiving a programmatic event corresponding to the action performed (*e.g., upon the occurrence of an event ... the clicking of a button icon ... invoke the action; col.13, lines 49-59*).

**As to claim 8:**

Strong teaches one or more parameters for specifying the default behavior of when the validation manager should validate text input for GUI elements described in the markup language file (*col. 6, line 22-col. 7, line 21 & see fig. 3B and the associated text*).

**As to claim 9:**

Strong teaches one or more attributes for specifying when text input to the GUI element should be validated; the validation manager is operable to override the default behavior and validate text input to the GUI element in accordance with the one or more attributes specifying when text input received to the GUI element should be validated (*col. 7, lines 5-41; and col.8, lines 1-64*).

**As to claim 10:**

Strong teaches said validation manager indicating that the text input received to the GUI element is invalid comprises the validation manager requesting the application to alter the visual appearance of the GUI element (*see Figs. 4 and 5 and the associated text*).

**As to claim 11:**

Strong teaches validation manager indicating that the text input received to the GUI element is invalid comprises the validation manager displaying an informational user interface window (*see Figs. 4 and 5 and the associated text*).

**As to claim 12:**

Strong teaches one or more attributes for controlling text input validation for the GUI element; wherein said step of processing a markup language file comprises the application constructing a document object representing the markup language file; wherein instantiating the validation manager comprises the application passing a reference to the document object to the validation manager; wherein, in response to being instantiated by and receiving the reference to the document object, the validation manager is operable to traverse the document object in order to discover the one or more attributes for controlling text input validation for the GUI element (*col.3, lines 11-47; col.4, line 63-col.5, line 22; col.7, lines 5-41; and col.8, lines 1-64*).

**As to claim 13:**

Strong teaches the one or more attributes for controlling text input validation for the GUI element include an attribute for specifying a type associated with the GUI element (*col.7, lines 5-41; and col.8, lines 1-64*).

**As to claim 14:**

Strong teaches the one or more attributes for controlling text input validation for the GUI element includes one or more attributes for specifying when text input to the GUI element should be validated (*col. 7, lines 5-41; and col. 8, lines 1-64*).

**As to claim 15:**

Strong teaches the one or more attributes for controlling text input validation for the GUI element include one or more attributes for specifying how invalid text input for the GUI element should be indicated (*e.g., if the data in the field is invalid according to requirements specified in the one or more configuration registry keys, an error message corresponding to the field being evaluated is dynamically built and logged in an error log; col. 3, lines 36-40*).

**As to claim 16:**

Lee teaches the validation manager component is a COM object (*See Figs. 6-13 and the associated text*).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify Strong with Lee because it would have provided the advantages including flexible and robust data validation, processing and error reporting for many different electronic forms in many different environments, as well as a

reduction in potential security risks that may be posed by providing data validation and processing information within an electronic form.

**As to claim 17:**

Lee teaches the validation manager component is a Java object (*See Figs. 6-13 and the associated text*).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify Strong with Lee because it would have provided the advantages including flexible and robust data validation, processing and error reporting for many different electronic forms in many different environments, as well as a reduction in potential security risks that may be posed by providing data validation and processing information within an electronic form.

**As to claim 18:**

Strong teaches the markup language is HTML (*e.g., Hypertext Markup Language; col. 3, lines 7-32*).

**As to claim 19:**

Strong teaches the type associated with the GUI element is a type comprising, among other things, date (*e.g., Input can only be a date; col. 8, lines 43-44*).

## Response to Arguments

3. Applicants' arguments filed 05/29/2007 have been fully considered but they are not persuasive.

Applicant argued in substance that *Strong's validation mechanism takes place on the server side, not on the client side.*

In response, the new combination of strong and Lee teaches validation mechanism takes place on the server side, not on the client side (*see Lee; col.5, line 23-col.6, line 39: the personal computer ...is operated by the user to control the validation rules program 15... the logic used by the validation rules program 15 to enable the user to create a set of validation rules for a particular form to be executed by a mobile pen application (MPA) on the mobile computer 30. The validation rules program 15 begins in a block 100 and proceeds to a block 110 in which the user selects a particular form for which the user wishes to create validation rules. As shown in more detail in FIG. 6, the validation rules program 15 generates a create validation rules window 400 on the display 14 of the personal computer 10 upon start-up. When the user selects a new rule file option 410 from the create validation rules window 400, the validation rules program 15 displays a listbox 420 containing the names of a number of forms available for validation rule*

*creation. A listbox is simply a display box having names of items which may be chosen listed therein. A user may also type in new items which will be included in the list).*

## **Conclusion**

4. The prior art made of record, listed on PTO 892 provided to Applicant is considered to have relevancy to the claimed invention. Applicant should review each identified reference carefully before responding to this office action to properly advance the case in light of the prior art.

## **Contact information**

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Maikhahan Nguyen whose telephone number is (571) 272-4093. The examiner can normally be reached on Monday - Friday from 9:00am – 5:30 pm. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Doug Hutton can be reached at (571) 272-4137.

The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/M. N./

*William L. Bashore*  
**WILLIAM BASHORE  
PRIMARY EXAMINER**